Software Requirements Specification

For

Email Spam Classifier

version 1.0 approved

Prepared by:

Namit Varshney
Registration No: 189302111

Mayank Jhanwar
Registration No: 189303179

Manipal University Jaipur

November 1, 2020

Table of Contents

Revision History:

Name	Date	Reason For Changes	Version

1 | Introduction

1.1 Purpose

Spamming is one of the major attacks that accumulate many compromised machines by sending unwanted messages, viruses, and phishing through E-Mails**.** It conveys the principal aim behind choosing this project under with. There are many people who try to fool you, just by sending you fake e-mails! Some of them have the messages like - You have won 1000 dollars, or we have deposited this much amount to your account. Once you open this link, then they will track you and try to hack your information.

The scope for this project is to identify all spam e-mails with the help of Machine Learning Algorithms.

1.2 Document Conventions

There are no used conventions till now.

1.3 Intended Audience and Reading Suggestions

We intend this document for anyone (such as developers, project managers, programmers, users, testers) who is interested in knowing about an E-Mail Spam Classifier.
The SRS contain answers for the following questions:

- **WHY an E-mail Spam Classifier is made?**
- **HOW an E-mail Spam Classifier is made?**
- **WHAT are the requirements for an E-mail Spam Classifier?**

1.4 Product Scope

Spamming is a big problem! Anyone who is connected to internet faces this problem now and then, but with Corporates, it becomes a serious problem. Phishing attacks, greedy traps and many create a havoc if not handled properly.
Our product **E-Mail Spam Classifier** is a solution for these Spams. Through our Intelligent Web- Based Product, you can find Spam E-Mails and can label them as Spam for future reference.

1.5 References

1.5.1 Websites

- www.kaggle.com
- www.sci-hub.tw
- www.ieeexplore.ieee.org
- www.medium.com

1.5.2 Dataset

- https://www.kaggle.com/venky73/spam-mails-dataset

2 | Overall Description

2.1 Product Perspective

The context of this product is as described above, and our team's goal is to make a safer and crime-free Cyber-World.
This is the first time we are working on this product, and it is something new for us. We are fully aware of the fact that there will be several challenges to make this product efficient and trustworthy. As spammers are getting smarter each day, we also need highly efficient Machine Learning Algorithm to tackle and knock out these Spammers.

2.2 Product Functions

Major highlights of our product's functionality are:

- This product will help **identify Spam E-Mails** similar to the Spam encountered earlier, which are stored in a vast library of Spam E-Mails.
- This product will also help in **identifying new Potential Spam E-Mails** from known & unknown sources. This what is going to be a speciality of our product.

2.3 User Classes and Characteristics

Our user class is vastly distributed across all kinds of Computer-related Industries. Some of them are mentioned below:

- **General Public**:
  Our Sole important target is common people who use E-Mail daily for their works and general internet communications. These are the one who gets the most E-Mail spams, and according to surveys, the public is the most favourite targets of these Spammers.
- **Corporate Companies**:
  Next big user class for this Product will be Corporate Companies which, because of employee's negligence and inadequate knowledge about Spam E-mail and trap their traps, makes companies' data vulnerable and pose great economic losses.

2.4 Operating Environment

The project is accessible from any web browser on a majority of devices.

2.5 Design and Implementation Constraints

Not all spam emails could be identified.

2.6 User Documentation

You can go to **https://emailspamclassifier.pythonanywhere.com/** and enter the Email or Message of the Email you want to check for spam in the space provided. Click on Check Spam or Not to start off the process.

2.7 Assumptions and Dependencies

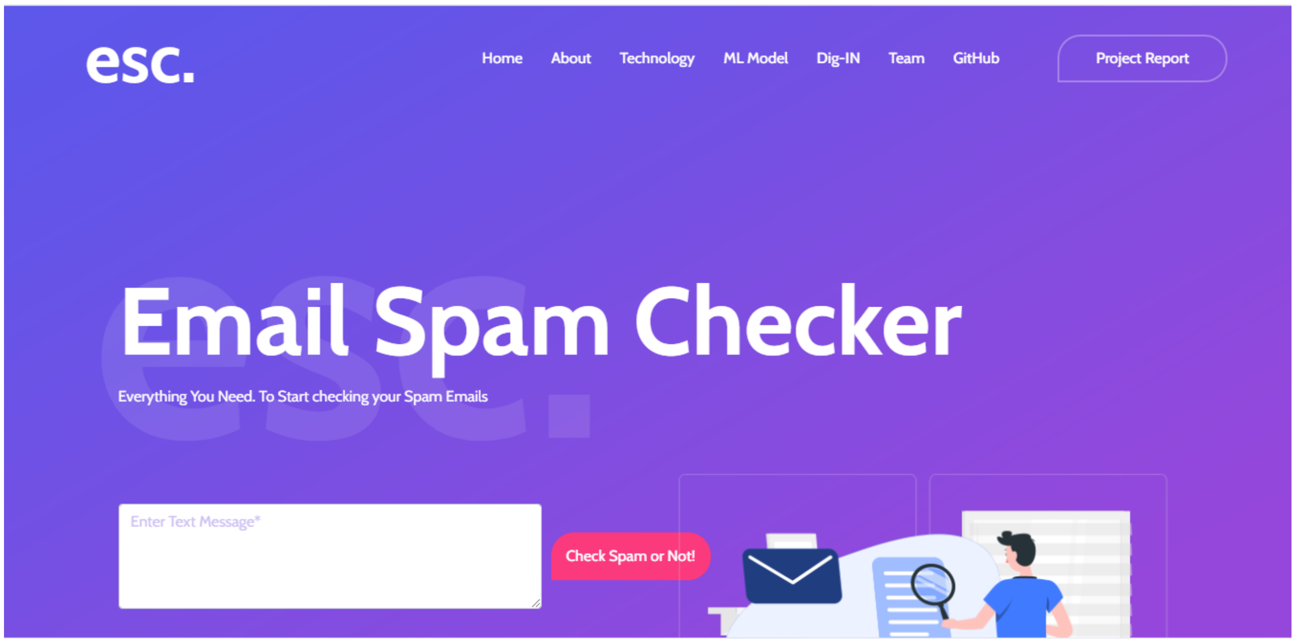
Our Software doesn't use any Third-Party tool component.

3 | External Interface Requirements

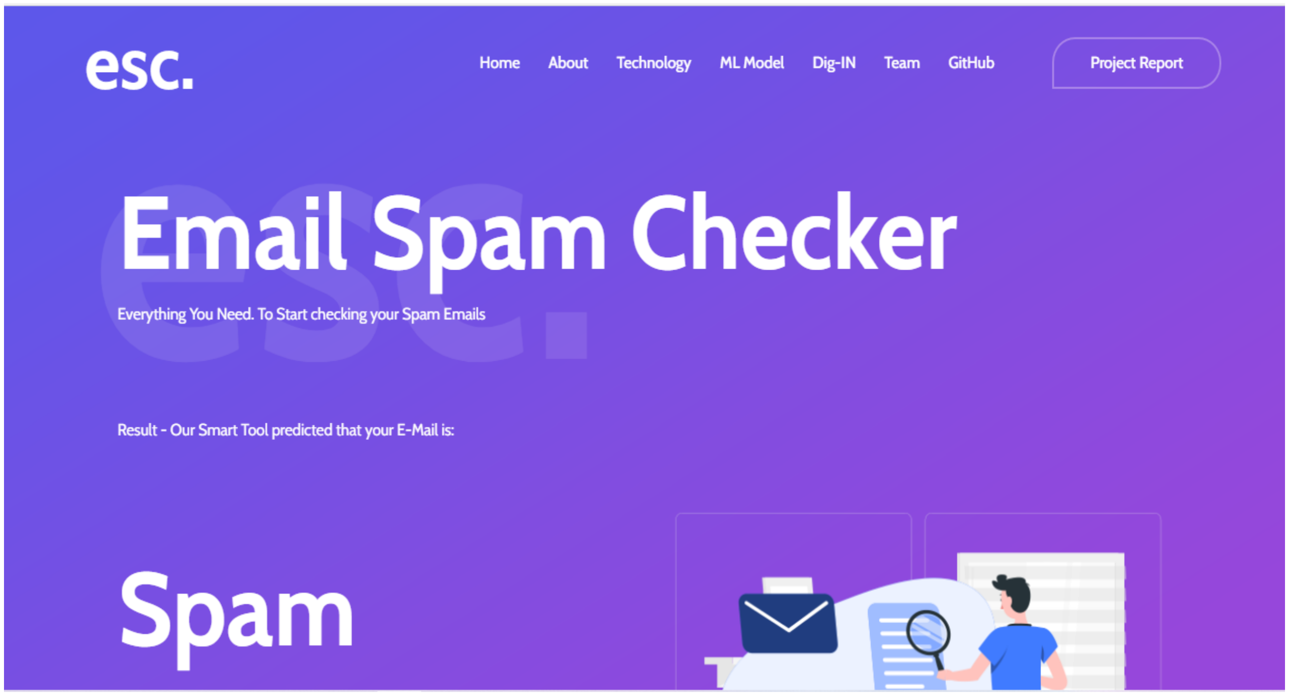
3.1 User Interface

The User Interface of our Project is simple, user-friendly, and self-explanatory. As per our current product Idea, we have a basic Website which is being hosted on the internet globally which has certain key options to check Spam E-Mails.

Opening Page @ https://emailspamclassifier.pythonanywhere.com/



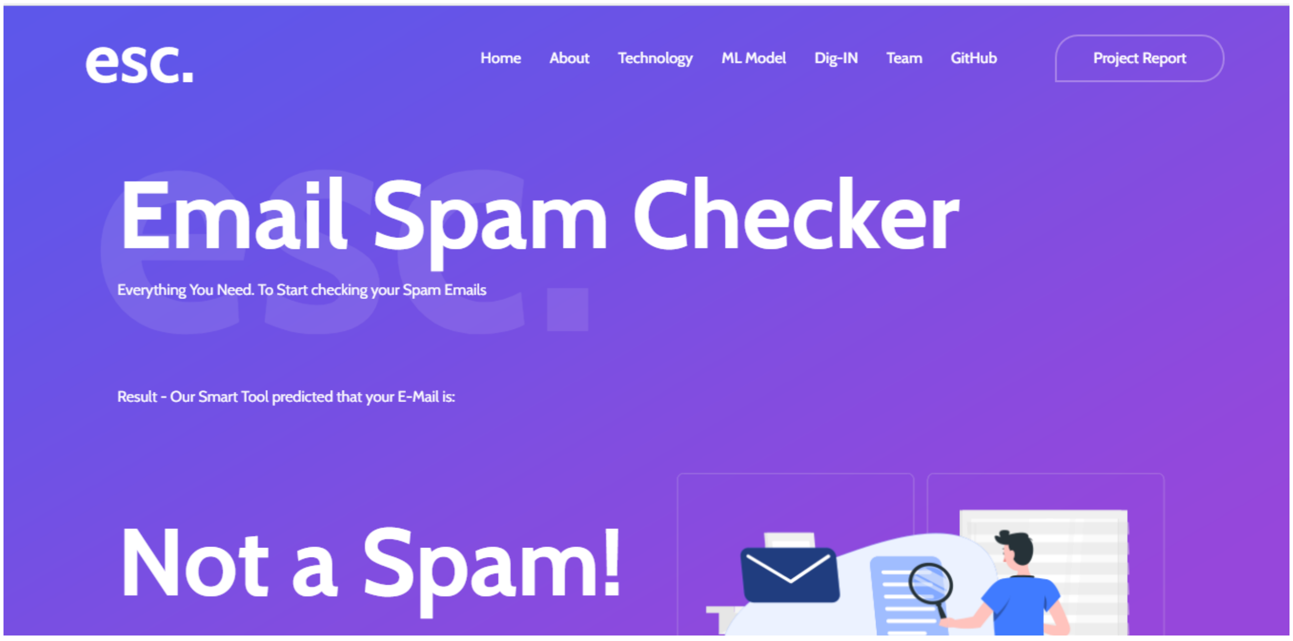
Results Page: If the Entered E-Mail is identified as Spam



Result - Our Smart Tool predicted that your E-Mail is:

Spam

Results Page: If the Entered E-Mail is identified as Not Spam



Email Spam Checker

Everything You Need. To Start checking your Spam Emails

Result - Our Smart Tool predicted that your E-Mail is:

Not a Spam!

3.2 Hardware Interfaces

The Hardware required to run this Product is very minimal and is mentioned below:

- **Operating System – Win/Mac/Linux (any distribution)**
- **Web Browser–Chrome/Mozilla/Opera/Tor (Updated with latest FTP/HTTP support)**
- **Proper Internet Connection (above 1 Mbps)**

3.3 Software Interfaces

This section discusses in detail, about how we made our Project Functional with Graphical User Interface :

- **Front-end:**
  The project's GUI has been made through web pages written in HTML and design elegantly with CSS and JavaScript for it to be attractive.

- **Back-end Framework:**
  Flask helps in implementing a machine learning application in Python that can be easily plugged, extended, and deployed as a web application.

- **Machine Learning Model:**
  The model detects, if a text message/mail is spam(1) or ham(0) using techniques like tokenization, multinomial naive bayes classifier, etc.

- **Web Server:**
  All these functional units are connected by establishing a server on the web by hosting it on Python Everywhere, which helps us in running the project successfully.

3.4 Software Interfaces

This project supports all types of web browsers

4 | System Features

4.1 Identification of Spam

The context of this product is as described below, and our team's goal is to make a safer and crime-free Cyber-World:

• Identify new Potential Spam E-Mails from known & unknown sources. This would our only priority
• The user will be asked to enter a message into the text field then the algorithm will detect whether it's spam or not.

This is the first time we are working on this product, and it is something new for us. We are fully aware of the fact that there will be several challenges to make this product efficient and trustworthy. As spammers are getting smarter each day, we also need highly efficient Machine Learning Algorithm to tackle and knock out these Spammers.

4.2 Machine Learning Algorithm

The usage of the libraries imported can be explained as:

- **NumPy** - for handling data arrays and making matrices.
- **Pandas** - for analysing and manipulating data.
- **Natural Language Toolkit(NLTK**) - for accessing text processing libraries like tokenization.
- **Stopwords** - removes useless words(data).
- **Strings** - for accessing string operations.

Functional Requirements

The main function of this project is to detect whether the given email is spam or not which is done by implementing certain algorithms such as :

- **Naïve-Bayes-Classifier:** Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

- **Tokenization**: Tokenization is the process of dividing text into a set of meaningful pieces. These pieces are called tokens. For example, we can divide a chunk of text into words, or we can divide it into sentences. Depending on the task at hand, we can define our own conditions to divide the input text into meaningful tokens.

5 | Non-Functional Requirements

5.1 Performance Requirements

The project is accessible from any web browser on a majority of devices.

5.2 Safety Requirements

We implement a variety of security measures to maintain the safety of your personal information when you enter, submit your email addresses to be checked.

5.3 Security Requirements

We do not sell, trade, or otherwise transfer to outside parties your personal information.

5.4 Software Quality Attributes

The software is using a large, labelled dataset of 5000+ messages tagged as spam or ham. So, there is very less probability that it could not identify a message

5.5 Business Rules

This Project is restricted within the college so no Business rules are required.

Appendix A : Glossary

4.1 List of Abbreviations

- **ML** – Machine Learning
- **AI** – Artificial Intelligence
- **DL** – Deep Learning
- **ISP** – Internet Service Provider
- **IP** – Internet Protocol
- **URL** - Uniform Resource Locator
- **IPV6/IPV4** - Internet Protocol version 6/ version 4
- **PHP** – Hypertext Pre-processor
- **HTML** – Hyper Text Mark-up Language
- **CSS** – Cascading Style Sheet
- **FTP** – File Transfer Protocol
- **HTTP** – Hyper Text Transfer Protocol